

通信制限のある複数エージェントの協調巡回清掃問題 における担当領域の重複とその抑制手法の提案

吉村 祐^{1,a)} 杉山 歩未¹ 菅原 俊治^{1,b)}

概要：本研究では通信に制限がある環境で、エージェント間の交渉を通じて自分が作業すべき担当領域を自律的に決定する手法を提案する。近年のロボット技術の発達により、ロボットの活躍の領域が広がっている。しかし環境の大きさ、求められる作業量などを考慮すると、複数のロボットによる協調作業が必要となる。我々は協調連続清掃問題を例題として、作業領域を分担しながら公平で効率的な作業のための環境の分割方法を提案してきた。しかし、通信範囲に制限がある環境では、隣接エージェントと長期間離れていて通信できない間に担当領域に冗長性が発生し、効率低下を招くことが分かった。そこで本論文では、各エージェントが隣接エージェントの作業負荷を推定しながら、拡大を制限する手法を提案し、評価実験によって手法の有効性を示す。既存手法で発生した不要な冗長性を防ぐことで効率を向上させただけでなく、必要な相手には助けるという形でこの冗長性を活用し、通信が常に可能な環境を想定した手法よりも効率化が実現できたことを実験を通して示す。

A Method for Preventing Redundant Responsible Areas Caused by Limited Communications in Multi-Agent Cleaning Task

YOSHIMURA YU^{1,a)} SUGIYAMA AYUMI¹ SUGAWARA TOSHIHARU^{1,b)}

1. はじめに

近年のロボット技術の発達により、ロボットの活躍の領域が広がっている。このような領域には、人間の負担を減らすための作業や人間が活動するのが困難な場面などがあり、たとえば、前者として清掃や警備、後者の例としては惑星探索や災害時の人命救助が挙げられる。このような応用分野では、継続的な探索や作業と、これらの行動を通してたとえば作業や探索の頻度の条件などを満たすように環境全体を訪れることが求められる。しかし、全ての作業を1台のロボットのみで処理するにはバッテリー容量や移動速度などの物理的あるいは能力上の制限があり現実的ではない。複数ロボットの協調作業の実現において、作業の継続性は必須な機能であり、ロボットの一部が故障しても協調

により相互に補い合い、作業を維持できるように、ロボットが自律的に行動を決定する必要がある。そのため本研究では、自律分散制御による協調作業に焦点を当てる。

複数のエージェントが協調して、継続的に環境を巡回するアプローチは大きく2つに分けられる。第1の手法は全エージェントが作業領域全体を共有して清掃する手法である [1-5]。たとえば、Sugiyama and Sugawara [4] では、[3] を拡張し、さらに分業を促進するための交渉手法を提案し、効率化を実現している。しかし領域を分割しない方法では、他のエージェントとの領域重複作業が起き、これだけでは十分な効率性を引き出せない可能性がある。第2のアプローチは領域を分割して、各エージェントが担当領域で作業する手法である [6-11]。たとえば、Sea et al. [9] では、ゴミの発生しやすさが未知の環境下で環境の差、エージェントの探索アルゴリズムの差、バッテリー容量の差などを反映した自律的分割手法を示した。しかし、この研究では隣接エージェントと常に通信が可能であることを仮定しており、状況によってはこの条件は厳しいこともある。

¹ 早稲田大学

Waseda University

a) y.yoshimura@isl.cs.waseda.ac.jp

b) sugawara@waseda.jp

本研究では、既存の手法 [9] に通信可能範囲の制限を加えても、作業効率を低下させない手法を提案する。

本稿の構成は以下の通りである。まず、第2節で本研究で用いるエージェントや環境のモデルを説明する。第3節で通信が不可能な間でも、不要な領域拡大をできる限り抑える領域分割法の提案をする。第4節では実験で用いるエージェントや環境について説明したのち、評価実験と考察を加える。最後に第5で結論と今後の課題を述べる。

2. モデルの定義

2.1 環境のモデル

$A = \{a_1, \dots, a_N\}$ をエージェントの集合とし、移動の表現のために離散時間を導入する。エージェントが清掃を行う空間を連結グラフ $G = (V, E)$ とする。ここで $V = \{v_1, \dots, v_x\}$ はノードの集合、 E はエッジの集合とし、ノード $v_i, v_j \in V$ にエッジが存在するときそれを $e_{i,j}$ と表し、 v_i, v_j 間に到達できる道があると考え、エッジの長さを全て1とし、エージェントは1単位時間でエッジで繋がった隣接ノードへと移動できる。

環境内のゴミの発生は単位時間ごとに確率的に起こり、各ノード v の汚れやすさを1単位時間あたりのゴミの発生確率 p_v ($0 \leq p_v \leq 1$) で表現する。時刻 t におけるノード v のゴミの量 $L_t(v)$ は、

$$L_{t+1}(v) = \begin{cases} L_t(v) + 1 & (\text{ゴミの発生時. 確率 } p_v) \\ L_t(v) & (\text{その他. 確率 } 1 - p_v) \end{cases} \quad (1)$$

のように蓄積されるが、エージェントが時刻 t に v に到達すると、 v 上のゴミは全て清掃され、 $L_{t+1}(v) = 0$ となる。

2.2 エージェントのモデル

エージェントのモデルとして、領域のマップを既知とする。また、エージェント a_i は有限容量のバッテリーを持ち、自分の基地から探索を開始し、自分の基地に戻り充電を行う。エージェントは探索状態と充電状態の2つの状態を交互に繰り返す。探索状態とは、エージェントが巡回清掃のために領域内を探索している状態のことである。エージェント a_i が持つバッテリーの最大容量を B_{max}^i 、1単位時間あたりのバッテリー消費量を B_{drain}^i とする。時刻 t におけるバッテリー残量を B_t^i とすると、時刻 $t+1$ におけるバッテリー残量は $B_{t+1}^i = B_t^i - B_{drain}^i$ となる。また、今回は通信によるバッテリー消費を考慮しない。エージェント a_i は基地 $v_{base}^i (\in V)$ で充電する。充電にかかる時間をバッテリーの消費量と比例すると仮定し、充電の比例定数を k_{charge}^i とすると、充電の所要時間 t_{charge}^i は $t_{charge}^i = k_{charge}^i (B_{max}^i - B_t^i)$ となる。

2.3 担当領域の同定とゴミの発生確率の学習

エージェント a_i には、初めに基地 v_{base}^i を中心に距離

d_{ini} ノード以下の範囲を初期担当領域 $G_0^i = (V_0^i, E_0^i)$ として与える。また、 $v_{base}^i \in V_t^i$ とする。以下、担当領域の同定とゴミの発生確率の学習について述べる。

エージェント a_i はノード v のゴミの発生確率 p_v を知らないため、 a_i は、担当領域内の各ノード v に対し、 p_v を学習する。エージェント a_i がノード v を時刻 t に訪れたとき、その前に v に訪れた時間を t_v^i とし (もし v が新しく担当領域になり、 a_i が v を訪れたことがなければ v が V_t^i に加えられた時刻を t_v^i とする)、ノード v を訪れた間隔 $t - t_v^i$ とノード v でのゴミの回収量 $AL_t(v)$ と学習率 α ($0 \leq \alpha \leq 1$) から v でのゴミの発生確率 $p_v^i(t)$ を以下のよう

$$p_v^i(t) = (1 - \alpha) \cdot p_v^i(t-1) + \alpha \cdot \frac{AL_t(v)}{t - t_v^i} \quad (2)$$

ここで、式 (2) は、観測値の差分を埋めることを示している。エージェントは時刻 t におけるノード v の実際のゴミの量 $L_t(v)$ を知らないが、上記で述べたように、ゴミの発生確率 p_v を学習するため、その値を使用した期待値として、ノード v におけるゴミの量の推定値 $E(L_t(v))$ を計算できる。エージェント a_i の担当領域におけるゴミの残量 $L_t(V_t^i)$ と a_i が推定する期待値 $E(L_t(V_t^i))$ は以下のように表せる。

$$L_t(V_t^i) = \sum_{v \in V_t^i} L_t(v) \quad (3)$$

$$E(L_t(V_t^i)) = \sum_{v \in V_t^i} E(L_t(v)) = \sum_{v \in V_t^i} p_v(t - t_v^i) \quad (4)$$

2.4 エージェントの行動

エージェントの行動は [10] に基づくものであり、ここではそれらについて簡単に述べる。

エージェントは探索アルゴリズムを使って担当領域を巡回する。本研究では有向深さ優先探索アルゴリズムを用いる。この探索アルゴリズムでは、エージェント a_i は基地から出発するとき担当領域の中からゴミの量の推定値 $E(L_t(v))$ の値が最も高いノードをターゲットノードとして最短距離で移動する。ターゲットノードに到達したら、隣接ノードのうち訪れていないノードをランダムに選択し、移動する。このプロセスをエージェント a_i が訪れたことがないノードを選べる限り続ける。この際、もしエージェント a_i は選択するノードがない場合、直前のノードに戻り、別の訪れていないノードを選択する。この動作を繰り返したのちターゲットノードへ戻った場合、基地に最短経路で戻る。

2.4.1 領域拡大行動

エージェントはある条件を満たしたとき担当領域をほとんど清掃したと判断し、担当領域の拡大を試みる。この判断のために、エージェント a_i は時刻 t_b に基地 v_{base}^i を出

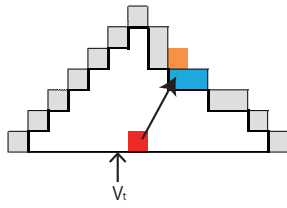


図 1: 拡大戦略

Fig. 1 Expansion strategy

発するとき、担当領域における少し未来のゴミの残量の推定値 $E(L_{t_b+\gamma}(V_{t_b}^i))$ を計算する。ここで γ は正の整数とする。これはエージェントが巡回清掃している間もゴミは蓄積し続けるため、もし現在を基準とすると、推定値よりも多くのゴミが残っているにも関わらず担当領域を十分に清掃したと判断し不要に領域拡大をする可能性があるからである。エージェント a_i は t_b から始まった現在の巡回行動中に自分が訪れたノード数 $N_{vis}(t)$ と回収したゴミの総和 $N_d(t)$ を記録する。そして、エージェントは、

$$N_{vis}(t) \geq R_1 \cdot |V_t^i| \quad (5)$$

$$N_d(t) \geq R_2 \cdot E(L_{t_b+\gamma}(V_{t_b}^i)) \quad (6)$$

のいずれかの条件を満たしたとき、担当領域を十分清掃できたと判断する。ここで、 $0 \leq R_1, R_2 \leq 1$ とする。なお、エージェントは十分なバッテリー残量を持っていても、領域拡大行動は基地を出てから1回のみとする。

拡大行動は、エージェントやシステム全体の性能に影響すると考えられる、基地からの距離や頻繁な拡大失敗の回避といったいくつかの要素を考慮して、拡大対象となるノード $I^i \subset V$ を決定する。まず、エージェント a_i は自分の担当領域の境界ノードの集合 B を選択する (B と担当領域は排反とする)。次に、エージェント a_i は B の中から、 I_{avoid}^i (これは後に説明する) に含まれず、かつ自分の基地 v_{base}^i から最も近い $k_{inc} (> 0)$ 個のノードの集合 I_{inc}^i とする。最後に、ノード I_{inc}^i とそれらの隣接ノードの集合のうち、 V_t^i にも I_{avoid}^i にも含まれないものを I^i とする。

例えば、図1に示すようなグリッド状の環境 G で、 V_t^i を太線内のノードの集合とし、境界ノード B を灰色と水色のノードとする (オレンジ色は含まれない)。 $k_{inc} = 1$ かつ $I_{avoid}^i = \emptyset$ 、 I_{inc}^i として矢印で指すノードを選んだとする。このとき水色とオレンジのノードが I^i となる。 $I^i = \emptyset$ となったときは a_i は領域拡大行動はしない。

重複領域の担当の決定や拡大戦略の指標として作業の余力を示す拡大力 (extension power) を定義する。エージェント a_i は時刻 t に充電基地に戻った際に、現在の担当領域に対する拡大力を計算するために、その時点でのゴミの残量の推定値 $E(L(V_t^i))$ を求め、エージェント a_i の拡大力 $\xi(i, t)$ を以下のように定義する。

$$\xi(i, t) = E(L(V_t^i))^{-1}$$

ただし $E(L(V_t^i)) = 0$ のときは $\xi(i, t)$ は十分大きな整数値とする。各エージェントはこの値を再び基地に戻り再計算するまで保持する。

2.4.2 通信と領域の更新 (領域拡大のための交渉)

エージェントは担当する領域の情報を定期的を送信し、隣接エージェントとの重複があるかを確認し、あった場合はどのエージェントが担当するかを決定するための交渉を行う。通信と交渉に関する部分は本稿の提案部分であり、次の節で説明する。交渉によって重複領域を担当領域から除外したエージェントは該当のノードを I_{avoid}^i として記録し、しばらくの間 (k_{avoid} 回分)、領域拡大行動の際には選ばないようにする (ただし、 k_{avoid} は正整数)。

2.5 システムの評価指標

本システムの評価は、ある期間 $[t_s, t_e]$ における全ノード V のゴミの存在量の総和の平均値 $D_{t_s, t_e}(V)$ とし、以下の式で表す。

$$D_{t_s, t_e}(V) = \sum_{t=t_s}^{t_e} L_t(V) / (t_e - t_s + 1) \quad (7)$$

この値をなるべく小さく保つことが本研究の目的である。また、全領域を均一に清潔に保つためには、各担当領域が均一に清掃されることが望ましい。そのため、 $D_{t_s, t_e}(V)$ に加え、各担当領域におけるゴミの存在量 (汚れの度合い) $L_t(V_t^i)$ の担当領域間の差も評価対象とする。

3. 提案手法

提案手法では、エージェント a_i は領域拡大にあたって、既存手法で定義した I_{avoid}^i ではなく、通信できない隣接エージェントの清掃状況を推定し、それに基づいた拡大戦略を用いる。

3.1 提案手法における通信と領域の更新

本研究では、エージェントはある一定の範囲内のエージェントとのみ通信できると想定する。エージェント a_i が送信した情報が届く通信範囲を距離 d_{comm} ($d_{comm} > 0$) 以下とする。本稿の提案手法における重複領域に関する交渉の流れは以下の通りである。

- (I) 情報の送信: 作業中のエージェントは、時間 N_{com} 毎にエージェント a_i は V_t^i と現在の拡大力 $\xi_i = \xi(i, t)$ を a_j に送信する。
- (II) 送信情報の受信: 時刻 t にエージェント a_j は、 a_i との距離が d_{comm} 以下のとき a_i と通信ができ、 a_i からの情報を受け取り、以下の動作を行う。
 - (1) 重複領域を $V_{overlap} = V_t^j \cap V_t^i$ とする。 $V_{overlap} = \emptyset$ の場合、何もせず終了する。 a_j は、 a_i の拡大力 ξ_i を記録する。
 - (2) $V_{overlap} \neq \emptyset$ なら、 a_j は、 a_i の拡大力 ξ_i と自分

表 1: 実験設定
Table 1 Experimental Setting

	パラメータ	値
エージェント台数	$ A $	4 台
学習率	α	5×10^{-2}
拡大戦略を行うためのパラメータ	R_1	0.7
	R_2	0.7
	γ	300
拡大戦略の制御パラメータ (既存手法のみ)	k_{inc}	15
	k_{avoid}	7
バッテリー消費量	B_{drain}	1
バッテリー最大容量	B_{max}	900
充電にかかる時間の比例定数	k_{charge}	3
通信が可能な範囲	d_{comm}	10
初期担当領域の範囲	d_{ini}	5
通信間隔	N_{com}	1

の拡大力 $\xi(j, t)$ を比較する。もし $\xi(j, t) \geq \xi_i$ ならば、 a_j は a_i に拡大拒否メッセージとして $V_{overlap}$ と $\xi_j = \xi(j, t)$, V_t^j を送信する。もし $\xi(j, t) < \xi_i$ ならば、 a_j は重複領域のノードを担当領域から削除し、 a_i に受諾メッセージとして $V_{overlap}$ と $\xi_j = \xi(j, t)$ および修正した V_t^j を送信する。

(III) a_i は a_j からの拡大拒否メッセージを受信した場合、そのノードを V_t^i から削除する。

なおこのとき、エージェント a_i と a_j は、情報交換した相手の拡大力 ξ_j と ξ_i を保存・更新し、さらに a_i は境界ノードの集合 B と a_j の担当領域 V_t^j の和集合 B_j^i を保存する。

3.2 提案手法における拡大戦略

提案手法における拡大戦略は、隣接エージェントの清掃状況を考慮して拡大するノードの集合 I^i を以下のように決定する。

- (1) すべての隣接エージェント a_j に関して、 $\xi(i, t)$ と保存した ξ_j を比較し、もし $\xi_j \geq \xi(i, t)$ ならば、 $B = B \setminus B_j^i$ とする。
- (2) a_i は B の中から自分の基地 v_{base}^i から最も近い順に k_{inc} 個のノードの集合である I_{inc}^i を選ぶ。
- (3) ノード I_{inc}^i とそれらの隣接ノードの集合のうち、 V_t^i と $\xi_j \geq \xi(i, t)$ を満たす B_j^i の両者に含まれないものを I^i とする。

I^i の要素をさしあたり担当領域 V_t^i に加え、清掃対象とする。

4. 評価実験

4.1 実験設定

提案手法の性能と特徴を実験を通して明らかにする。実験で用いるエージェントと環境の設定を表 1 に示す。4 台のエージェント $A = \{a_1, a_2, a_3, a_4\}$ はすべて同じ性能とす

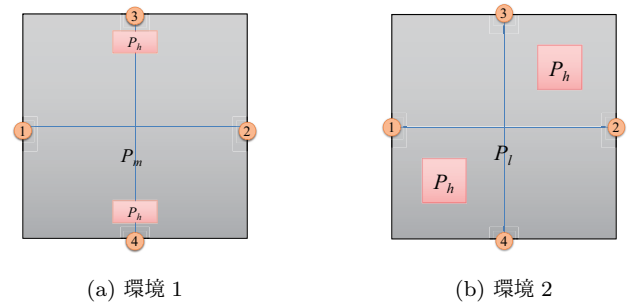


図 2: 実験環境

Fig. 2 Experimental environments

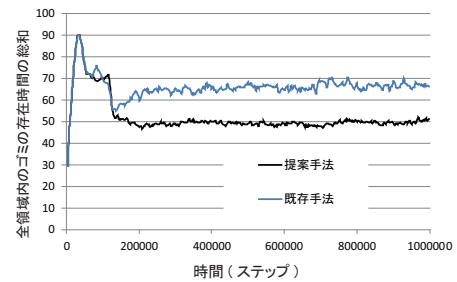


図 3: 全ノードのゴミの存在時間の総和の推移 (環境 1)

Fig. 3 Accumulating existence duration of dirt (environment 1)

る。本実験の環境は図 2 に示すように、エージェントが清掃を行う空間 G を縦横共に 51 の 2 次元グリッドとする。エージェント a_i は各々の充電基地 v_{base}^i (図 2 では丸囲み数字で表す) から各担当領域 V_t^i の巡回清掃を行う。

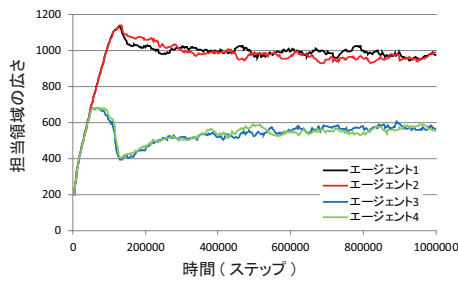
環境に汚れやすさの偏りを持たせるために、ゴミの発生確率として $p_h = 2 \times 10^{-4}$, $p_l = 2 \times 10^{-6}$ の 3 つがあり、図 2 のように配置した。環境 2 は担当領域の境界と思われる付近に発生確率の高い領域があり、他方、環境 1 では基地に近い部分の発生確率が高くなっている。実験は 1 回 100 万単位時間とし、3600 単位時間ごとに以下の値を記録する。

- (1) 全ノードのゴミの存在時間の総和 $D_{t_e, t_s}(V)$ (以下 $D(V)$ と略記する)
- (2) 各担当領域の広さ $|V_t^i|$
- (3) 各担当領域におけるゴミの残量 $D(V_t^i)$

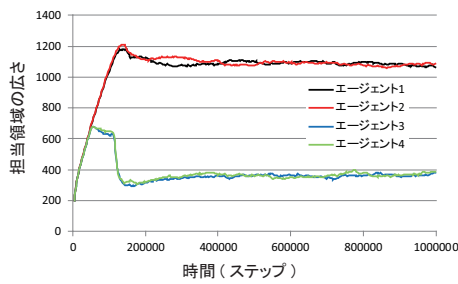
比較手法として [10] で提案した既存手法を採用する。既存手法では、第 2.4.1 節で述べた拡大戦略で担当領域を拡大し、次の通信可能なタイミングまでその領域も清掃を続ける。

4.2 実験結果

環境 1 と環境 2 をまとめて比較すると、その結果に大きな差があることが分かる。図 3 は、環境 1 における $D(V)$ の推移を、図 4 に、 $|V_t^i|$ の推移を示す。同様に環境 2 の $D(V)$ の推移を図 5 に、 $|V_t^i|$ の推移を図 6 に示す。



(a) 既存手法



(b) 提案手法

図 4: 担当領域の広さの推移 (環境 1)

Fig. 4 The sizes of the responsible area (environment 1)

表 2: 環境全体のゴミの残量の分散

Table 2 Variance of remaining dirt in each environment

環境	手法	分散値
環境 1	既存手法	84.98
	提案手法	36.81
環境 2	既存手法	13.83
	提案手法	8.49

図 3 から環境 1 においては、既存手法と比べて提案手法が $D(V)$ を減少させている (減少量は約 15%)。他方、図 5 から環境 2 において、 $D(V)$ はほぼ同じである。また、図 4(a) (b) を比較すると、環境 1 の各担当領域の広さに関しては、提案手法のエージェントは既存手法と比較して、エージェント 3 と 4 が担当領域を大きく減少し、エージェント 1 と 2 は増加している。また、各環境における重複領域の大きさを図 7 に示す。

4.3 考察

ここで提案手法が既存手法と比べて効率を向上させた要因を考察する。分析の結果、効率向上の要因には大きく 2 つあり、それぞれの詳細を説明する。

第 1 の要因 (要因 1) は、重複領域の減少である。重複があると各担当領域が広くなり、無駄な冗長性が生まれ回収効率が低下する。実際に、図 7 に示すように、既存手法より担当領域の重複を低下させた提案手法は既存手法と少なくとも同等、あるいは明らかに清掃効率を向上させている。

第 2 の要因 (要因 2) は、提案手法による領域拡大の際に負荷の高い方向へのみ拡大することにより、暗黙的に負荷

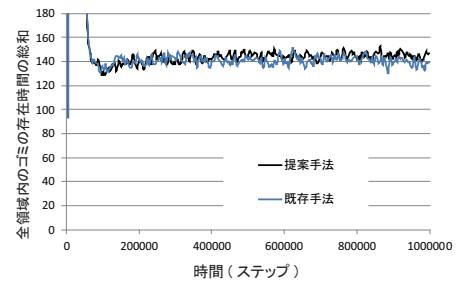
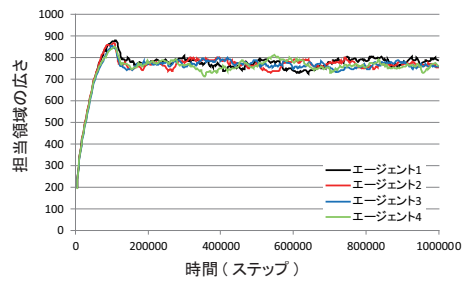
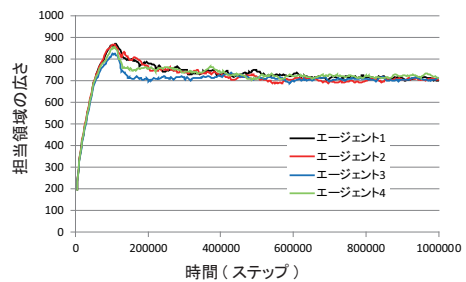


図 5: 全ノードのゴミの存在時間の総和の推移 (環境 2)
Fig. 5 Accumulating existence duration of dirt (environment 2)



(a) 既存手法



(b) 提案手法

図 6: 各担当領域の広さの推移 (環境 2)

Fig. 6 The sizes of the responsible area (environment 2)

の高いエージェントを助ける「適度な相互援助」の行動である。効率的な巡回のためには、領域分割によって各エージェントに負荷が適切に分担されることが望ましい。負荷の分散度合いをみるために、表 2 に既存手法と提案手法それぞれの、エージェント毎のゴミの残量の分散値を示す。この表から、どちらの環境においても提案手法はゴミの残量の分散値が小さく、平均として環境を綺麗に保つだけではなく、巡回問題において重要な評価指標である均一的な清掃もできていることが分かる。

分散値を低下させた理由は、環境 1 のように汚れやすい領域に偏りがあり、担当領域のサイズに差が発生した場合は明らかである。提案手法では通信で得た近隣の拡大力の通信時点での値 ξ_j と自己の現在時刻での拡大力 $\xi(i, t)$ を比較する。したがって、提案手法でも担当領域の重なりは発生するが、それは常に自分より負荷が高いと推定した方向のみに向かう。さらに担当領域拡大により自己の拡大力も下がるため、通信が長期間できなくともある時点で拡大行

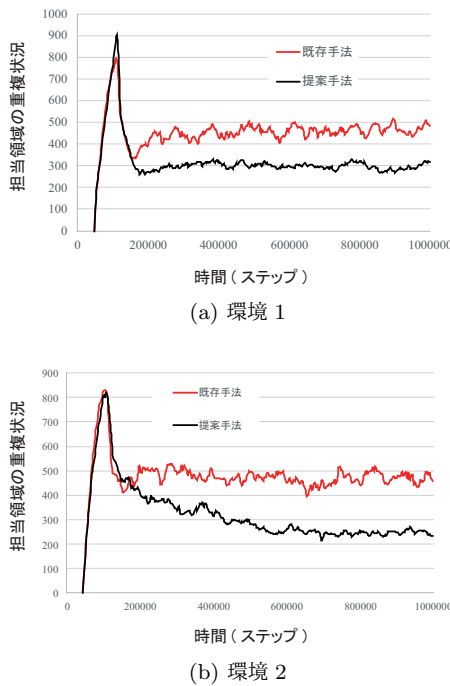


図 7: 各環境における重複領域の大きさ

Fig. 7 the size of overlap area in environment 1,2

動は抑制される。これにより、ゴミの残量の分散値も減少する。つまり提案手法では、既存手法に比べ不要な重複領域を減少させただけでなく、重複領域は発生させるものの、それが負荷の高いエージェントを助けるように働く。領域拡張行動は拡大力の差に基づくため、その差が無くなったときは拡大を止める。結果として、提案手法は拡大行動による重複は防がないが、適切な方向への適度な拡大行動により、効率化が達成できたと考えられる。

5. まとめ

本研究では、領域分割を用いたマルチエージェント巡回清掃において、通信範囲の制限によって生じる冗長性が引き起こす清掃効率の低下を回避する手法を提案した。既存研究 [10] から、通信ができる機会が限られている環境では、通信による交渉だけでは重複領域を防ぐのには間に合わず、その環境全体の清掃効率の低下につながっていた。そこで本研究では、通信を行った際に隣接エージェントの情報を記録し、次に通信が可能となるまでの間、他者の清掃状況や負荷度合いを推定し、それを考慮しながら領域拡大を行うことで、整合性のある分業を行う手法を提案した。評価実験の結果、一部の負担の大きいエージェントの担当領域の過剰な拡張を防いだことで従来手法に比べて効率的かつ、より均等な領域分割が実現でき、提案手法の有用性を示せた。

今後の課題としては、エージェントと環境の大規模化が考えられる。領域分割法は隣接したエージェント同士がやり取りをしていくものであり、別のエージェントを仲介す

る必要があると考えられる。

参考文献

- [1] Othmani-Guibourg, M., Fallah-Seghrouchni, A. E., Farges, J. L. and Potop-Butucaru, M.: Multi-agent patrolling in dynamic environments, *2017 IEEE International Conference on Agents (ICA)*, pp. 72–77 (online), DOI: 10.1109/AGENTS.2017.8015305 (2017).
- [2] Chevaleyre, Y.: Theoretical Analysis of the Multi-agent Patrolling Problem, *Proceedings of Intelligent Agent Technology*, pp. 302–308 (2005).
- [3] Yoneda, K., Sugiyama, A., Kato, C. and Sugawara, T.: Learning and relearning of target decision strategies in continuous coordinated cleaning tasks with shallow coordination, *Web Intelligence*, Vol. 13, No. 4, pp. 279–294 (online), DOI: <http://dx.doi.org/10.3233/WEB-150326> (2015).
- [4] Sugiyama, A. and Sugawara, T.: Improvement of Robustness to Environmental Changes by Autonomous Divisional Cooperation in Multi-agent Cooperative Patrol Problem, *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection* (Demazeau, Y., Davidsson, P., Bajo, J. and Vale, Z., eds.), Cham, Springer International Publishing, pp. 259–271 (2017).
- [5] Sampaio, P. A., Ramalho, G. and Tedesco, P.: The Gravitational Strategy for the Timed Patrolling, *Proceedings of the 2010 22Nd IEEE International Conference on Tools with Artificial Intelligence - Volume 01, ICTAI '10*, Washington, DC, USA, IEEE Computer Society, pp. 113–120 (online), DOI: 10.1109/ICTAI.2010.24 (2010).
- [6] Nasir, A., Salam, Y. and Saleem, D. Y.: Multi-Level Decision Making in Hierarchical Multi-agent Robotic Search Teams, Vol. 1 (2009).
- [7] Wiandt, B., Simon, V. and Kokuti, A.: Self-organized graph partitioning approach for multi-agent patrolling in generic graphs, *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, pp. 605–610 (online), DOI: 10.1109/EUROCON.2017.8011183 (2017).
- [8] Kato, C. and Sugawara, T.: Decentralized Area Partitioning for a Cooperative Cleaning Task, *Proceedings of the 16th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA-2013)*, pp. 470–477 (2013).
- [9] Sea, V., Kato, C. and Sugawara, T.: Coordinated Area Partitioning Method by Autonomous Agents for Continuous Cooperative Tasks, *Journal of Information Processing*, Vol. 25, pp. 75–87 (online), DOI: 10.2197/ip-sjip.25.75 (2017).
- [10] 吉村 祐, 杉山 歩未, 菅原 俊治: 通信範囲に制限を持つマルチロボットの巡回清掃における効率的な領域分割法の提案, 電子情報通信学会技術研究報告 (人工知能と知識処理研究会), Vol. 115, No. 478, No. 478, pp. 7–12 (2016).
- [11] Jain, U., Tiwari, R., Majumdar, S. and Sharma, S.: Multi Robot Area Exploration Using Circle Partitioning Method, *International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)*, Procedia Engineering, Vol. 41, pp. 383 – 387 (online), DOI: <https://doi.org/10.1016/j.proeng.2012.07.188> (2012).